REMARKS

Claims 16-21 and 27-44 were rejected. Claims 16, 27, 29, 34, 42 and 44 have been amended. Claims 16-21 and 27-44 remain pending in the present application.

In a response, dated June 11, 2002, to a restriction requirement under 35 U.S.C. § 121, dated May 22, 2002, applicants elected to prosecute the claims designated in Group II, consisting of claims 16-21 and 27-44. In the June 11, 2002 response, applicants agreed to cancel the claims of Groups I and III without prejudice to reserve the right to file them in a later application. Applicants reiterate this response, and herein formally cancel claims 11

The office action rejected claims -15 (Group I) and claims 22-26 (Group III). 16, 18-20, 27, 28, 42 and 43 under 35 U.S.C. § 102 (a) as being anticipated by Baxter et al. (U.S. Patent No. 6,289,500) ("Baxter").

In reply to the applicants' previous response, the office action notes that "the claim language does not require a separate step of determining/considering whether a requested functionality is inherent in the class object prior to looking." (*Office Action dated 3/27/03* at p. 6). Applicants have amended the claims of the present invention to make explicit that which previously was implicit in the claims; namely, that the present invention determines whether a requested functionality is inherent in the class object. Therefore, applicants have amended the claims per the Examiner's suggestion and have not raised a new issue.

Applicants respectfully request the Examiner enter the amendments as offered. Specifically, applicants submit that the present amendments should be entered as per

M.P.E.P. § 713.14, as the amendments provide a basis to avoid the rejections set forth in the current official action placing the case in condition for allowance ***or in the alternative a better condition for appeal***. Also, the amended claims do not raise the issue of new matter, nor do the amendments present new issues requiring further consideration or search.

The office action further suggests that "in Baxter, by definition, domain-neutral extensible items/objects do not provide domain-specific functions" and therefore "Baxter considers the inherency in the class object by determining that a requested function is a domain-specific function/extension before locating/creating such function/extension." (*Office Action dated 3/27/03* at p. 6). Although the office action's characterization of Baxter in this regard may be correct, it misses the point. In particular, the present invention not only considers whether a requested functionality is inherent in the class object, but when such functionality is not inherent in the class object, the present invention creates an extension object ***from an extension package***. In Baxter, when a requested domain-specific function is not available, Baxter creates a new domain extension (*Baxter* - Figure 8 – step 810). However, there is no teaching or suggestion in Baxter that such an extension is created from an existing extension package, as claimed in the present invention. One example of such use of an existing extension package in the present invention occurs when a previously created extension object from one vendor's application is made available to another vendor's application.

Next, in reply to the applicants' previous response, the office action suggests that the present specification does not disclose or claim the above described example

– *i.e.*, that "extension objects from one vendor's application [will] be available to another vendor's application." (*Office Action dated 3/27/03* at p. 7). With all due respect to the contentions in the office action, the Examiner is respectfully requested to acknowledge that the present specification clearly discloses this feature as one of the many advantages of the claimed function and structure. In particular, the present specification clearly discloses that "because the extensible object model provides extensions that are accessible through late or ID binding and thus the extension objects do not have to be present when an application is developed, independent software vendors (ISVs) can readily extend an application from another vendor by proffering additional methods and/or properties for the objects comprising the application." (*Specification* – page 13, lines 21-23). This feature of the present invention further is described in even greater detail with reference to Figure 4 of the present specification. Furthermore, this feature results from the claimed recitation that "the extension package proffers an extension provider object when the functionality is requested."

As previously discussed, this aspect of the present invention is wholly different than Baxter. Baxter is directed to the "San Francisco" architecture, which is well known to those skilled in the art. In particular, Baxter does not teach or suggest extending the functionality of a class object by creating an extension object from an extension package when a requested functionality is not inherent in the class object. Instead, as clearly discussed with reference to Baxter's Figure 8, Baxter is directed to first creating a domain extension, finding its proper collection, and then creating an extensible item in that proper collection. This is consistent with Baxter's stated

motivation to "provide[] frameworks that define the basis of an application such as a general ledger or order management with well-defined extension points . . . [to provide] user-defined extensions that ***customize San Francisco for a particular application***." (*Baxter* – column 5, lines 51-56) (emphasis added). In other words, Baxter is directed to creating an architecture (*i.e.*, the "San Francisco" architecture) that is amenable to customization so that users can customize the architecture to fit their application.

The Examiner is respectfully asked to recognize the fundamental difference between Baxter's customizable architecture and the present invention which permits extension objects from one vendor's application to be to be available to another vendor's application by creating the extension object from the extension package. This aspect is a result of the present invention's ability to allow "the extensible object model to provide extensions that . . . do not have to be present when an application is developed."

Accordingly, applicants respectfully request withdrawal of the rejection under 35 U.S.C. § 102 (a) of claims 16, 18-20, 27, 28, 42 and 43 over Baxter.

The office action also further rejected claim 34 under 35 U.S.C. § 102 (a) as being anticipated by Graser *et al.* (U.S. Patent 6,275,979). Also, the office action rejected claims 17, 29-33, 35-41and 44 under 35 U.S.C. § 103 (a) as being unpatentable over the IBM San Francisco framework as disclosed by Baxter in combination with Graser, and rejected claim 21 under 35 U.S.C. § 103 (a) as being unpatentable over Baxter in view of Schmidt *et al.* "An Object-Oriented Framework for Developing Network Server Daemons" ("Schmidt").

For the same reasons discussed above with reference to the rejection of claims 16, 18-20, 27, 28, 42 and 43 under 35 U.S.C. § 102 (a) over Baxter, applicants respectfully request withdrawal of the rejection of claim 34 under 35 U.S.C. § 102 (a) over Graser, claims 17, 29-33, 35-41 and 44 under 35 U.S.C. § 103 (a) over Baxter in combination with Graser, and claim 21 under 35 U.S.C. § 103 (a) over Baxter in combination with Schmidt.

CONCLUSION

In view of the foregoing amendments and remarks, applicants respectfully submit that the present application is in condition for allowance. Reconsideration of the application and an early Notice of Allowance are respectfully requested. Also, in the alternative, the Examiner is respectfully requested to enter the amendment to put the claims in a better condition for appeal. In the event that the Examiner cannot allow the present application or enter the present amendments for any reason, the Examiner is encouraged to contact applicants' attorney Vincent J. Roccia at (215) 564-8946.

Respectfully submitted,

Date: **June 26, 2003**

Vincent J. Roccia
Registration No. **43,887**

WOODCOCK WASHBURN
One Liberty Place - 46th Floor
Philadelphia, PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

Marked up versions of claims 16, 27, 29, 34, 42 and 44 amended herein, showing all of the changes relative to the previous version of each.

Please cancel claims 11-15 and 22-26.

16. A system for extending functionality of a class object, comprising:

a processing unit;

a system memory in communication with the processing unit via a system bus;

a computer-readable medium in communication with the processing unit via the system bus; and

an extensible object model executed from the computer-readable medium by the processing unit, wherein the extensible object model determines whether a requested functionality is inherent in the class object, and wherein the extensible object model causes the processing unit to create an extension object from an extension package when a requested functionality is not inherent in the class object, and wherein the extension object extends the class object to provide the requested functionality.

27. A computer-readable medium having stored thereon computer-executable components comprising:

an extensible object;

an extension database having an entry for an extension for the extensible object; and

an extension package having an interface for obtaining an extension object that provides the extension for the extensible object, wherein the extensible package determines whether a requested functionality is inherent in the class object.

29. A method for extending functionality of a class object in a run-time environment, comprising:

determining whether a requested functionality is inherent in the class object;

receiving a request from an application for functionality that is not inherent in the class object;

determining if the functionality is available in a first extension object;

obtaining an extension package having computer-executable instructions associated with the extension object functionality, wherein the extension package proffers an extension provider object when the functionality is requested;

specifying parameters to the extension provider object to create a second extension object; and

directing the request to the second extension object.

34. A method for extending functionality of a class object in a run-time environment, comprising:

determining whether a requested functionality is inherent in the class object;

receiving a request from an application for functionality that is not inherent in the class object;

determining if the functionality is available in a first extension object; and

directing the request to the functionality in a second extension object, when the functionality is not available in the first extension object.

42. A system for extending functionality of a class object, comprising:

a processing unit;

a system memory in communication with the processing unit via a system bus;

a computer-readable medium in communication with the processing unit via the system bus;

an extensible object model executed from the computer-readable medium by the processing unit, wherein the extensible object model determines whether a requested functionality is inherent in the class object, and wherein the extensible object model creates an extension object from an extension package when a requested functionality is not inherent in the class object, and wherein the extension object extends the class object to provide the requested functionality.

44. A method for extending functionality of a class object, comprising:

determining whether a requested functionality is inherent in the class object;

invoking a functionality that is not inherent in the class object;

determining if the invoked functionality is available in a first extension object;

creating a second extension object when the invoked functionality is not available in the first extension object; and

directing the invocation to the second extension object.